Use of Semantic Tools for a Digital Rights Dictionary

Jaime Delgado, Isabel Gallego, Roberto García

Universitat Pompeu Fabra (UPF), Departament de Tecnologia,
Pg. Circumval·lació 8, E-08003 Barcelona, Spain
{jaime.delgado, isabel.gallego, roberto.garcia}@upf.edu

Abstract. RDDOnto is an ontology that translates the MPEG-21 RDD (Rights Data Dictionary) specification into a hierarchical set of definitions with semantic content included. In the event that this set of definitions is used, the RDDOnto must provide well-defined semantics to determine which rights apply to data at all points within the hierarchy. RDDOnto translates the RDD specification into a machine-readable semantic engine that enables automatic handling of rights expressions. The Terms defined in the RDD Specification are what is going to be modelled using OWL (Web Ontology Language). For each Term, its description is composed by a set of descriptive attributes. With OWL, all the RDD relations between a term and other terms that capture its semantics have been mapped to RDDOnto. The specification of MPEG-21 RDD using OWL has also allowed to verify the consistency of the dictionary. [1]

1 Introduction

One of the main problems of the electronic commercialisation of multimedia resources in Internet is the management of its associated digital rights. New solutions are required for the access, delivery, management and protection processes of different content types in an integrated and harmonised way, to be implemented in a manner that is entirely transparent to the many different users of multimedia services.

MPEG-21 Part 6, Rights Data Dictionary (RDD), comprises a set of clear, consistent, structured, integrated and uniquely identified Terms to support the MPEG-21 Rights Expression Language. In turn, MPEG-21 Part 5, Rights Expression Language (REL), defines a language that enables to declare rights and permissions using the terms as defined in the Rights Data Dictionary.

The objective of this work is to translate the RDD terms descriptions from its current textual representation in the RDD specification document [1] to a machine processable representation. Translating these descriptions to a machine-aware form would facilitate the integration of the RDD with the other parts of MPEG-21, specially REL, and the implementation of MPEG-21 compliant software tools.

In order to achieve these objectives, the target has been a knowledge representation framework with a wide range of utilities available. Our approach has been to use the

[1] This work has been partly supported by the Spanish Ministry of Science and Technology under the AgentWeb project (TIC2002-01336).

Semantic Web paradigm. The web-orientation of this approach would also facilitate the integration of MPEG-21 implementations in the World Wide Web scenario.

The Semantic Web paradigm is an attempt to leverage the Web from a distributed information repository to a distributed knowledge one. The Semantic Web basic tools are the Resource Description Framework (RDF) [2] and RDF Schema [3]. A more advanced tool is the Web Ontology Language (OWL) [4]. Using these tools and starting from our previous experience in developing a general ontology for Digital Rights Management [5], we have developed this ontology for the MPEG-21 RDD, IPROnto.

2 MPEG-21 Rights Data Dictionary (RDD)

The aim of MPEG-21 [6] is to define a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities. MPEG-21 is organized into several parts already developed or currently under development, see Table 1.

Table 1. MPEG-21 stantdard parts

Part 1: Vision, Technologies and Strategy	Part 8: Reference Software
Part 2: Digital Item Declaration (DID)	Part 9: File Format
Part 3: Digital Item Identification (DII)	Part 10: Digital Item Processing
Part 4: Intellectual Property Management and Protection (IPMP)	Part 11: Evaluation Methods for Persistent Association Technologies
Part 5: Rights Expression Language (REL)	Part 12: Test Bed for MPEG-21 Resource Delivery
Part 6: Rights Data Dictionary (RDD)	Part 13: Scalable Video Coding
Part 7: Digital Item Adaptation (DIA)	Part 14: Conformance Testing

The sixth part of MPEG-21 specifies a Rights Data Dictionary for use within the MPEG-21 Framework. This Rights Data Dictionary forms the basis of all expressions of rights and permissions as defined by the MPEG-21 Rights Expression Language. MPEG-21 sees a Rights Data Dictionary as a dictionary of key terms which are required to describe rights of all users, which can be unambiguously expressed using a standard syntactic convention, and which can be applied across all domains in which rights need to be expressed. A Rights Expression Language is seen as a machine-readable language that can declare rights and permissions using the terms as defined in the Rights Data Dictionary.

The RDD System comprises the RDD Dictionary (Terms and their TermAttributes) and the RDD Database (the tool containing the RDD Dictionary). The Rights Data Dictionary consists in a set of clear, consistent, structured, integrated and uniquely identified Terms to support the MPEG-21 Rights Expression Language. The StandardizedTerms are specifically defined to support the REL and provide the foundation of the RDD Dictionary. New Terms, developed specifically to support REL requirements, independently or from mappings from other schemes, can be added to the RDD Dictionary through the registration of such Terms with the

Registration Authority. The process to create such a Registration Authority to administer the RDD is under way.

As a closed ontology, all RDD terms are defined with reference to other Terms in the dictionary. This has two main consequences for the understanding of a term when it is used in an REL license. The first is that no assumptions should be made about the meaning of a term based on the coincidence that it bears the same name as something in an application domain. The second consequence concerns the inheritance of meaning. As the rights data dictionary is a hierarchical ontology, most of the meaning of a term is inherited from its parent(s) (in RDD terminology, its *Archetypes*). This RDD standard contains all the RDD StandardizedTerms listed in alphabetic order, each term is shown with its TermAttributes and all of its immediate Types and AllowedValues.

The Dictionary has the characteristics of a structured ontology, in which meaning, once it has been defined, can be passed on from one term to another by logical rules of association such as inheritance and opposition.

The fourteen *ActTypes* which provide basic functionality for the REL are: *Adapt, Delete, Diminish, Embed, Enhance, Enlarge, Execute, Install, Modify, Move, Play, Print, Reduce* and *Uninstall*. They are employed within a rights expression.These Multimedia Extension Rights are capable of being used to create licenses required by Rights Holders. The fourteen *ActTypes* have been defined in response to requirements identified in the process of developing the REL and RDD Standards, particularly focused on common processes in the use and adaptation of Digital Resources. However, it is recognised that in future further *ActTypes* will have to be introduced into the RDD Dictionary in response to new requirements from REL users.

3 Semantic Web concepts

In this section, RDF, RDFS and OWL are presented. RDF and RDFS are referred together as RDF/S. RDF is used to associate metadata to resources in order to make information about them explicit. Resources are named using URIs, i.e. URLs or URNs. The RDF modelling primitive is the graph. It is composed by a set of arcs used to assert property values about resources and to relate resources between them. Arcs are also called triples in RDF terminology. Each graph arc is composed by a subject URI (the resource about which the statement is made), a property URI and a value (literal) or an object URI (the resource to which the subject is related by the property).

An RDF description is composed by a set of arcs describing some resources. The set of arcs constitutes a graph that can be navigated in order to retrieve the desired metadata. There is an example in Fig. 1.

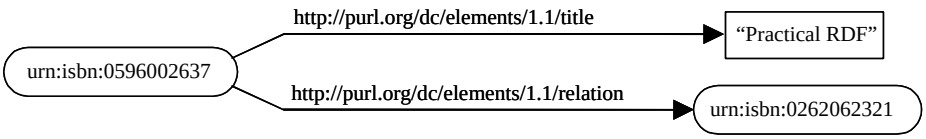

Fig. 1. RDF Graph constituted by three triples

As it has been seen until now, RDF provides a framework to model metadata. The basic primitive is the graph. This can be compared with the XML context, where the modelling tool is the tree. However, as an XML tree, an RDF graph is on its own basically unrestricted. Therefore, in order to capture the semantics of a particular domain, some primitives to build concrete "how things are connected" restrictions are necessary.

The tool that provides these restriction-building primitives is RDF Schema. It can be compared to XML Schema or DTDs, which provide building blocks to define restrictions about how XML elements and attributes are related. The primitives are some restricted URN names defined in the RDF and RDFS namespaces.

RDFS provides Object Orientation-like primitives. With these primitives, class hierarchies can be defined. Resources are declared members of some of these classes and inherit their associated restrictions. The RDF/S classes are summarised in Section 6.1 of the RDF Schema specification [7]. Moreover, there is a special kind of class: Property. It contains all the resources used to relate subject and object in triples. Property hierarchies can also be defined, and domain (origin) and range (destination) of the RDF graph arcs can be restricted to specific classes. They are summarised in Section 6.2 of the RDF Schema specification [8].

The Web Ontology Language is a more advanced ontology building toolkit. It provides more fine-grained primitives that allow additional restrictions. OWL will allow mapping almost all the relations in Genealogy to RDDOnto. Many of the RDD relations in Genealogy are not mappable using only RDF/S constructs, more details about that are presented in the next section.

OWL is superset of RDF/S, i.e. in an OWL ontology all the primitives of RDF/S can be used. Therefore, when we refer to OWL primitives, all the primitives from RDF/S will be also considered. The primitives are summarised in Appendix A of the OWL Web Ontology Language Reference [9].

4 RDDOnto: An Ontology for Rights Data Dictionary

The set of all predefined classes and properties are the building blocks provided by the OWL and RDF/S frameworks. These building blocks are used to construct Semantic Web ontologies, i.e. sets of restrictions to the basic RDF elements. These restrictions can be automatically validated in order to test that a particular RDF description conforms to the semantics of the particular domain captured by the ontology.

In the next subsection, we will detail how first RDF/S and afterwards OWL frameworks can be used to capture the definition of RDD terms and a great part of their semantics. RDF/S is capable of modelling only a fraction of the RDD semantics. This fraction is augmented when the constructs introduced by OWL are also used. Therefore, two versions of the ontology can be produced. The simpler one uses RDF/S and the more complex uses OWL.

4.1 RDD Specification analysis

The RDD Specification defines a set of terms. Terms are what is going to be modelled using RDF/S. For each term, its description is composed by a set of descriptive attributes:

- *Headword*: The term name. It must appear in the term description.
- *Synonym*: Some alternative names. It is not mandatory.
- *Definition*: A short text that defines the term.
- *MeaningType*: From a set of predefined types: *Original*, *PartlyDerived* and *Derived*.
- *Comments*: Extended textual information about the term. It is not mandatory.
- *Relationships*: Relations, from a set of predefined ones, between these terms an other terms that capture its semantics. The relations are classified in these categories:
  - *Genealogy*: The relations in this category will be the focus of RDDOnto. They are: *IsTypeOf, IsA, Is, IsEquivalentTo, IsOpposedTo, IsPartOf, IsAllowedValueOf, HasDomain, HasRange* and *IsReciprocalOf*.
  - *Types, Family, ContextView* and *Membership of Sets:* These categories will be analysed in future versions of RDDOnto. They are primarily concerned with the generative semantics of the RDD terms and they are less relevant during final ontology use.

These are the target attributes of RDDOnto. Their values will be mapped to OWL representation tools, which include also RDF/S ones, in order to capture the greatest part of their implicit semantics.

4.2 RDD to RDF/S mapping

From the RDD Specification analysis two kinds of attributes can be detected. The first group is composed by those attributes with unstructured values, i.e. textual values. They can be easily mapped to predefined or new RDF properties with textual (literal) values. The first option is to try to find predefined RDF properties that have the same meaning that the RDD term attributes that are being mapped. When this is not possible, the RDFS constructs will be used to define new RDF properties to which the corresponding attributes will be mapped.

The mappings of this kind are shown in Table 2. Note that the Dublin Core [10] RDF Schema is also reused in RDDOnto. The Dublin Core (DC) metadata element set is a standard for cross-domain information resource description. The DC RDF Schema implements the Dublin Core standard.

Table 2. RDF mappings for the RDD attributes with textual value

RDD Attribute	RDF Property	Kind of RDF property
Headword	rdf:ID	Predefined in RDF
Synonym	rddo:synonym	New property defined in RDDOnto
Definition	dc:description	Predefined in Dublin Core RDF Schema
MeaningType	rddo:meaningType	New property defined in RDDOnto
Comments	rdfs:comment	Predefined in RDFS Schema

The other kind of attribute is the Relationships one. Its value is not textual. Firstly, it is categorised in five groups: *Genealogy*, *Types*, *Family*, *ContextView* and *Membership of Sets*. Each of these groups is composed by a set of relation that can be used to describe a term related to other terms of the RDD Specification.

As has been justified in the previous section, only the Genealogy group is considered. The relations in this group are presented in the upper part of Table 3 together with a short description and the equivalent RDF property used to map them in RDDOnto. Only the RDD relations with an equivalent property in RDF/S are mapped at this level, i.e. *IsTypeOf*, *IsA*, *HasDomain* and *HasRange*. The other relations have associated semantics that do not have an equivalence in RDF/S.

Therefore, if the mapping is restricted to the possibilities provided by RDF/S, then we get an uncomplete ontology, i.e. it does not capture all the available semantics of RDD. However, on top of RDF/S, more advanced restriction building tools, like OWL, have been developed. In the next sections the improvements that can be done using OWL are presented.

4.3 RDD to OWL mapping

Using OWL ontology building blocks, some of the previously unmapped RDD relations can be mapped to the RDD ontology. In bottom part of Table 3 they are presented together with a short description and the equivalent OWL property used to map them in RDDOnto. With OWL all the RDD relations that have been considered relevant have been mapped to RDDOnto, i.e. except those from *Relationships* that are not the *Genealogy* group.

Table 3. RDF and OWL mappings for the RDD relations in the Genealogy group

RDD relation	Short description	RDF
IsTypeOf	Builds the hierarchy of term types	rdfs:subClassOf rdfs:subPropertyOf
IsA	Relates an instance term to its type	rdf:type
HasDomain	For relation terms defines the source term type of the relation	rdf:domain
HasRange	For relation terms defines the target term type of the relation	rdf:range

RDD relation	Short description	OWL
Is	Relates QualifiedResources to AscribedQualities	rddo:hasQuality
IsEquivalentTo	Relates two equivalent terms	owl:equivalentClass owl:equivalentProperty owl:sameIndividualAs
IsOpposedTo	Relates two opposite terms	owl:disjointWith (owl:complementOf)
IsPartOf	Relates a terms that is part of another term	rddo:isPartOf
IsAllowedValueOf	Relates an instance terms that is allowed value of a type term	Inverse of owl:oneOf
IsReciprocalOf	For relation terms defines the relation terms that captures the inverse relation	owl:inverseOf

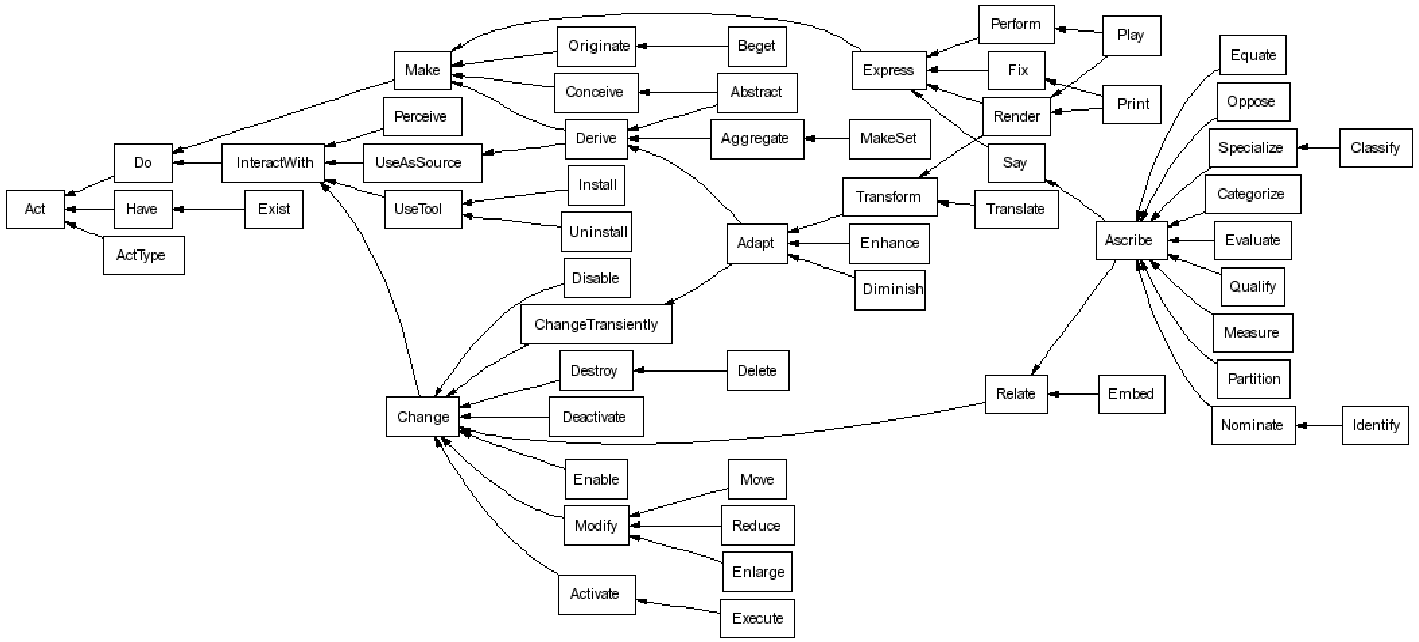

Fig. 2. ActType hierarchy diagram

4.4 Implementation

The RDD to RDF/S and OWL mappings that have been established in Table 2, Table 3 and Table 4 have been implemented in the RDDOntoParser. It is a Java implementation of these mapping using regular expressions [11]. Regular expressions are used to define patterns that detect the RDD part of the mappings. When patterns match, the corresponding RDF is generated in order to build RDDOnto.

The input of the RDDOntoParser is a plain text version of the Table 3 – Standardized Terms of the RDD Specification [1]. The output constitutes the RDDOnto Web ontology [12]. Fig. 2 shows a drawing of the Act hierarchy generated automatically from RDDOnto using the Protégé [13] ontology editor with the OntoViz visualisation plug-in.

During the mapping process, some inconsistencies in the RDD specification have been found. One group are inconsistencies between RDD terms definitions and their graphical representations. Another group are references to terms not defined in the specification.

5 Using RDDOnto

Once the RDD ontology is ready, the Semantic Web tools that are available can be used to develop MPEG-21 RDD implementations. Some of these tools are presented in [14] and in [15]. In the next subsection, our experience with one of these tools is presented.

5.1 Using RDDOnto with Sesame

Sesame [16] is an RDF tool. Concretely, it is an RDF repository. It is used to store both RDF Schemas and metadata. In other words, it can store together web ontologies (Classes and Properties definitions) and instances of them that constitute resource descriptions. The stored schemas and metadata can be queried using three different query languages, navigated or serialised to RDF.

A Sesame RDF repository containing RDDOnto can be accessed at [17]. Sesame repositories can be easily created installing the Sesame software over a Java servlet container (e.g. Tomcat) and then configuring a metadata repository using a relational database (MySQL, Postgres or Oracle).

Once the Sesame repository is ready, the Sesame tool can be accessed using a web browser. From this interface, RDDOnto can be interactively uploaded and queried. For a programming interface, the RDDOntoAPI has been implemented. Some details are presented in the next section.

5.2 Using RDDOnto from Java

An RDDOntoAPI has been developed with Java in order to interact with RDDOnto, once it has been loaded into Sesame. This API is used to integrate RDDOnto with other tools, such as our REL (MPEG-21 Rights Expresssion Language) License Interpreter [18].

In order to facilitate API integration, RDDOntoAPI is also available from a web service interface. Web services are specified using the Web Services Description Language (WSDL) [19]. The RDDOntoAPI WSDL specification is available from [20]. Table 5 shows and example of use of the web service interface.

Table 5. Instancing the client service proxy to invoke getRDDSuperTypes operation

```
import org.systinet.wasp.webservice.Registry;
import edu.upf.dmag.mpegontos.iface.RDDOntoAPI;
…
String wsdlURI = http://hayek.upf.es:8080/wasp/MPEGOntosAPI";
RDDOntoAPI service =
 (RDDOntoAPI)Registry.lookup(wsdlURI, RDDOntoAPI.class);
String[] superTypes = service.getRDDSuperTypes("Play");
```

The getRDDSuperTypes operation is used to retrieve from the RDD ontology (RDDOnto) all the parents of the given type, from the RDD point of view. For instance, if the RDD term is an act type like *Play*, all the parent act types will be returned: *Transform*, *Render*, *Perform*, *UseAsSource*, *Make*, *InteractWith*, *Expres*,…

This operation is used during license checking. If it fails for the required right, the parents of the right are retrieved and checked as, from the semantics of the acts hierarchy, it is derived that they include the rights appearing as subtypes.

It is implemented as a query submitted to the Sesame repository using one of the Sesame's query languages, RQL [21]. It is an augmented version of SQL that allows exploiting the greater expressive power of RDF, compared to relational databases.

6 Conclusions and Future Work

We have presented RDDOnto, an ontology for the MPEG-21 Rights Data Dictionary (RDD). Its added value over other initiatives to implement rights data dictionaries is that it is based on applying an ontological approach. This is done by modelling the RDD standard using ontologies. Ontologies allow that a greater part of the standard is formalised and thus more easily available for implementation, verification, consistency checking, etc.

RDDOnto demonstrates the benefits of capturing the RDD semantics in a computer-aware formalisation. It can be seen that it is easier to integrate RDD in order to develop MPEG-21 tools.

MPEG-21 tool implementers can use the API in order to facilitate access to many characteristics of this standard that are quite inaccessible from the resources directly provided, i.e. informal terms specifications. Indeed, this has been proven during the development of the REL License Interpreter [18].

Future plans are focused on applying also the ontological approach to another part of the MPEG-21 specification: the Rights Expression Language (REL). REL is specified in MPEG-21 using a different approach: XML Schemas.

Our intention is to automate the XML schema mapping to OWL ontology step. Then, once the REL ontology is available, it would be easier to integrate it with RDD as both will be formalised using the same language, OWL.

References

1. ISO/IEC FDIS 21000-6, MPEG-21 Rights Data Dictionary (RDD). N5842, July 2003
2. Lassila, O. and Swick, R.R. (eds.): "Resource Description Framework (RDF), Syntax Specification". W3C Recommendation, 2004, http://www.w3.org/TR/rdf-syntax-grammar
3. Brickley, D. and Guha, R.V. (eds.): "RDF Vocabulary Description Language 1.0: RDF Schema". W3C Recommendation, 2004, http://www.w3.org/TR/rdf-schema
4. Dean, M. and Schreiber, G. (eds.): "OWL Web Ontology Language Reference". W3C Recommendation, 2004, http://www.w3.org/TR/owl-ref
5. Delgado, J, Gallego I., Llorente, S., García R.: Regulatory Ontologies: An Intellectual Property Rights Approach. LNCS, Vol. 2889. Springer-Verlag, 2003. 621-634
6. Moving Picture Experts Group (MPEG) ISO/IEC/ JTC1 SC29/WG11, http://www.chiariglione.org/mpeg/index.htm
7. RDFSchema Summary: Classes, http://www.w3.org/TR/2004/REC-rdf-schema-20040210/#ch_sumclasses
8. RDFSchema Summary: Classes, http://www.w3.org/TR/2004/REC-rdf-schema-20040210/#ch_sumproperties
9. OWL Web Ontology Language Reference: Appendix A. Index of all language elements, http://www.w3.org/TR/2004/REC-owl-ref-20040210/#appA
10. Dublin Core Metadata Element Set, Version 1.1: Reference Description, http://dublincore.org/documents/dces/
11. Java 2 Platform Std. Ed. v1.4.2, Package java.util.regex, http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/package-summary.html
12. RDDOnto (RDD Ontology), http://dmag.upf.edu/ontologies/2003/11/rddonto.owl
13. Protégé Project, http://protege.stanford.edu
14. Dave Beckett's Resource Description Framework (RDF) Resource Guide http://www.ilrt.bris.ac.uk/discovery/rdf/resources/#sec-tools
15. DAML Tools, http://www.daml.org/tools
16. Sesame RDF Repository, http://sesame.aidministrator.nl
17. DMAG Sesame Repository, http://hayek.upf.es/sesame
18. Rodríguez, E.; Delgado, J. and Llorente, S.: "DMAG REL license interpretation using RDD term genealogy". ISO/IECJTC1/SC29/WG11/M10287. December 2003
19. Web Service Definition Language (WSDL), www.w3.org/TR/wsdl
20. MPEGOntosAPI Web Service Description Language (WSDL) specification, http://hayek.upf.es:8080/wasp/MPEGOntosAPI
21. RQL (Rdf Query Language) Tutorial, http://www.openrdf.org/doc/rql-tutorial.html